# Bowdoin Computer Science

# Reasons to study Computer Science

- Computing is part of everything we do!
- Expertise in computing enables you to solve complex problems
- Computing enables you to make a positive difference in the world
- Computing offers many types of careers
- Computing jobs are here to stay

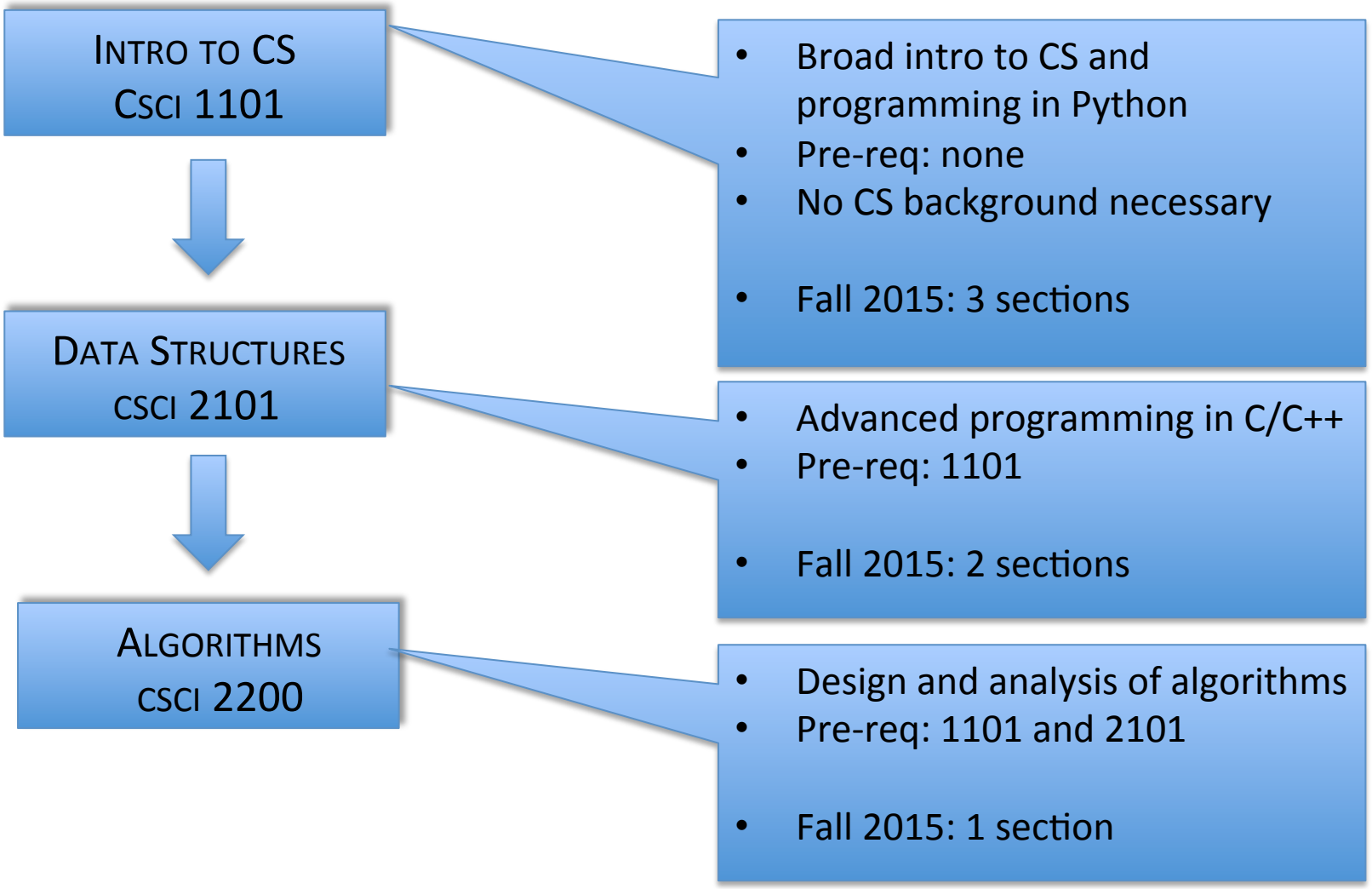
From http://computingcareers.acm.org/

# Reasons to study Computer Science

- Knowledge of computing helps you even if your primary career choice is something else
- Computing offers great opportunities for creativity
- Some knowledge of computing is becoming a sign of well-roundedness
- Future possibilities in computing are without boundaries

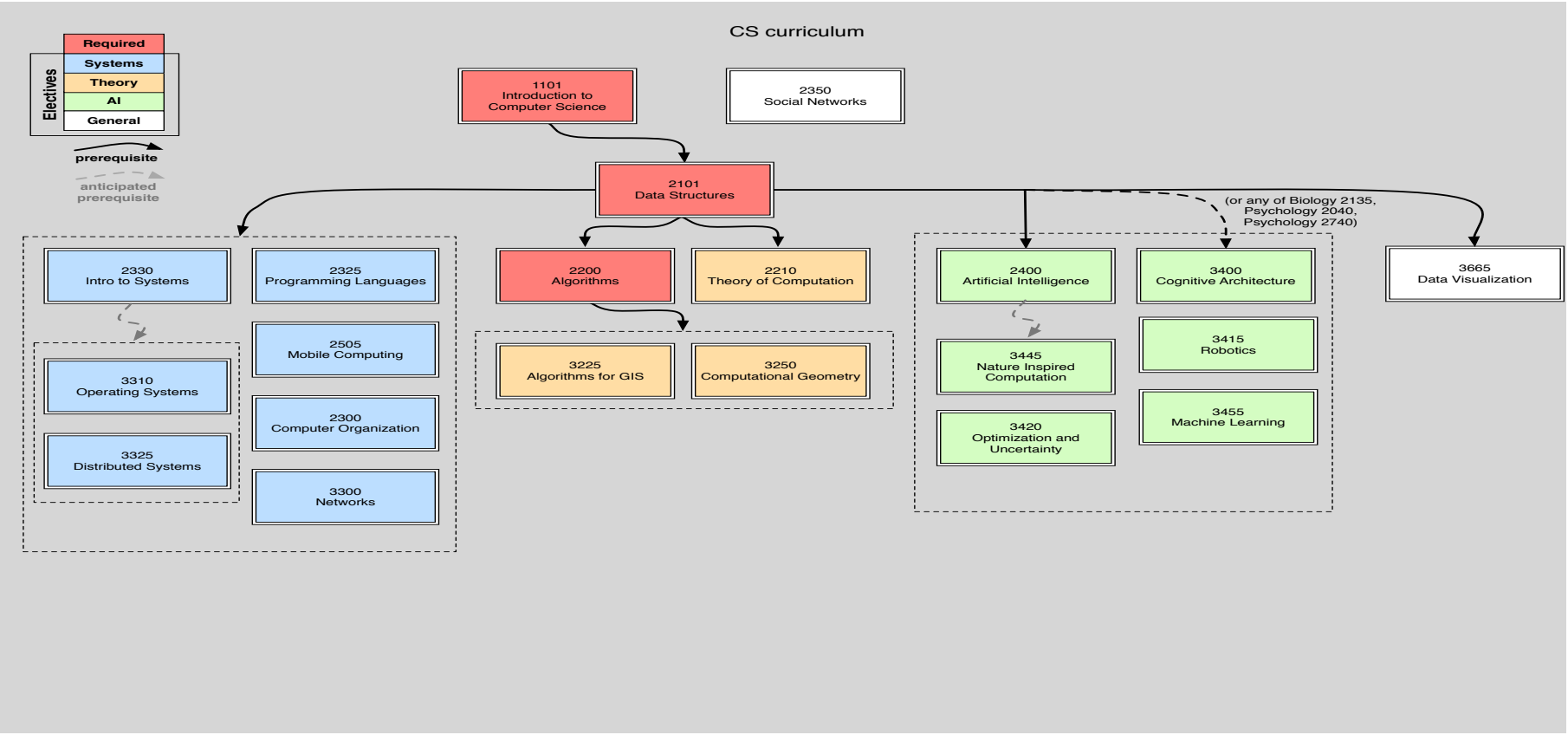
From http://computingcareers.acm.org/

# The CS intro sequence



# The CS Major

10 classes: 1101, 2101, 2200 + 7 electives



The CS minor:

5 CS classes: 1101, 2101 + 3 electives

The interdisciplinary Math-CS major:

CS: 1101, 2101, 2200 + 3 electives
Math:…

# Entry-level computing-related classes

## Intro to CS
CSCI 1101

- Introduction to problem solving using computer programming

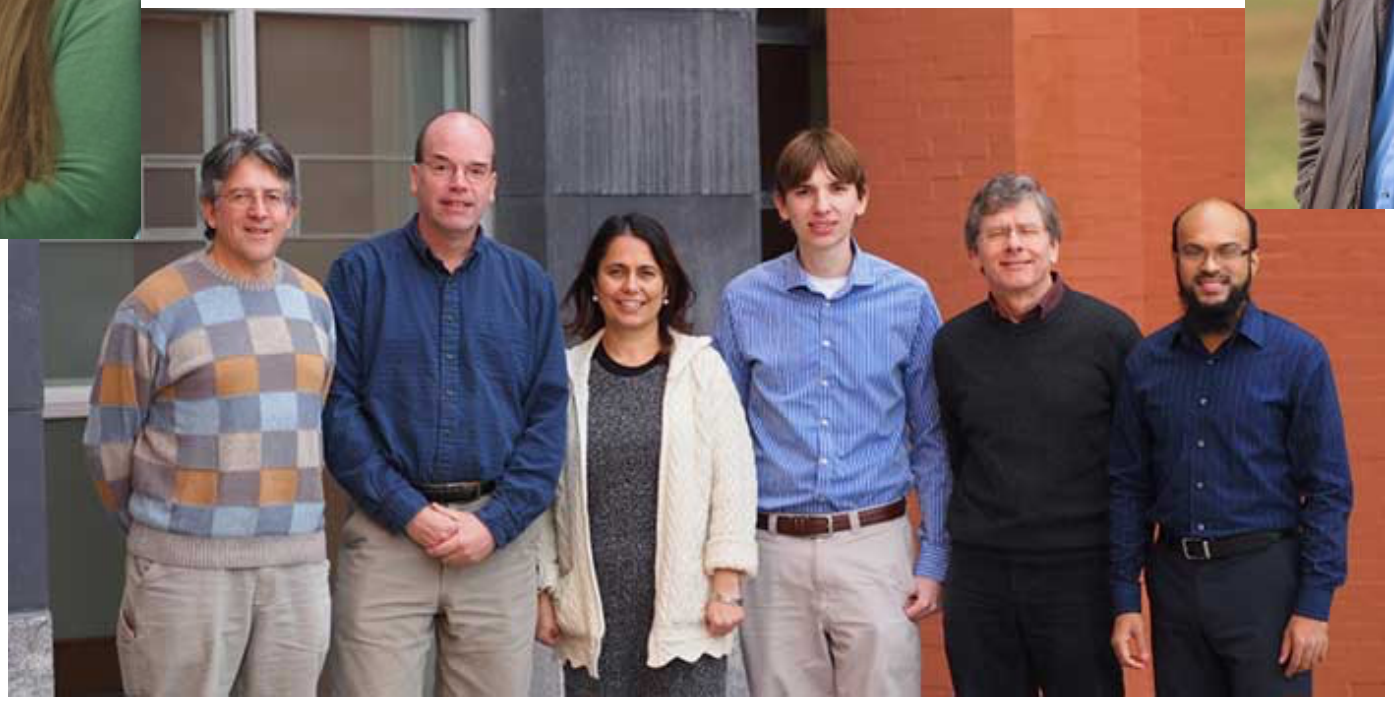
Required for CS major/minor

## Intro to DCS
INTD 1100

- Study of values, behavior and technologies associated with digital environments
- Includes some Python programming and web apps

Both offered in Fall'16

# CS Faculty



Sean Barker

Eric Chown

Clare Congdon (visiting)

Allan Harper (visiting)

Mohammad Irfan (joint with DCS)

Steve Majercik

Laura Toma

# CS Research

- Independent studies
- Honors thesis
- Summer research

# CS Research Areas

Sean Barker

Distributed systems, cloud computing, sustainability

Projects

- Designing sustainable smart homes through analysis of smart meter data
- Resource management in data centers (ie memory sharing in virtual machines)
- On-demand live migration in cloud-based databases

# ReCS Research Area

Eric Chown

Cognitive modeling, soccer-playing robots

Projects

- Bowdoin's NorthernBites team competing in RoboCup

# CS Research Areas

Clare Congdon

Machine learning, bioinformatics

Projects

- Find patterns in noncoding DNA sequence that appear to have been conserved across evolutionary time
- Find the most plausible evolutionary relationships among species
- Virtual Simulation of the Lobster Fishing Industry in the Gulf of Maine

# CS Research Areas

Allen Harper

Human-computer interaction, eye tracking

Projects

- Predict how well a user performs a task based on eye movement. Classify users into performance groups.

# CS Research Areas

Mohammad Irfan

Computational game theory, social and economic networks, CS and art

Projects

- Modeling influence in economic networks
- Analyzing Kandinsky's art through geometric primitives
- Authentication of Jackson Pollock's paintings

# CS Research Areas

Steve Majercik


Nature-inspired computational techniques, swarm intelligence and particle swarm optimization, computation and the arts.

Projects

- Jazz improvisation tool using particle swarm optimization (with Frank Mauceri, Music Dept)

- Swarm-based path creation in dynamic environments for search and rescue

# CS Research Areas

Laura Toma

Efficient algorithms for large data; terrain processing in GIS; algorithm engineering; high-performance computing

Projects

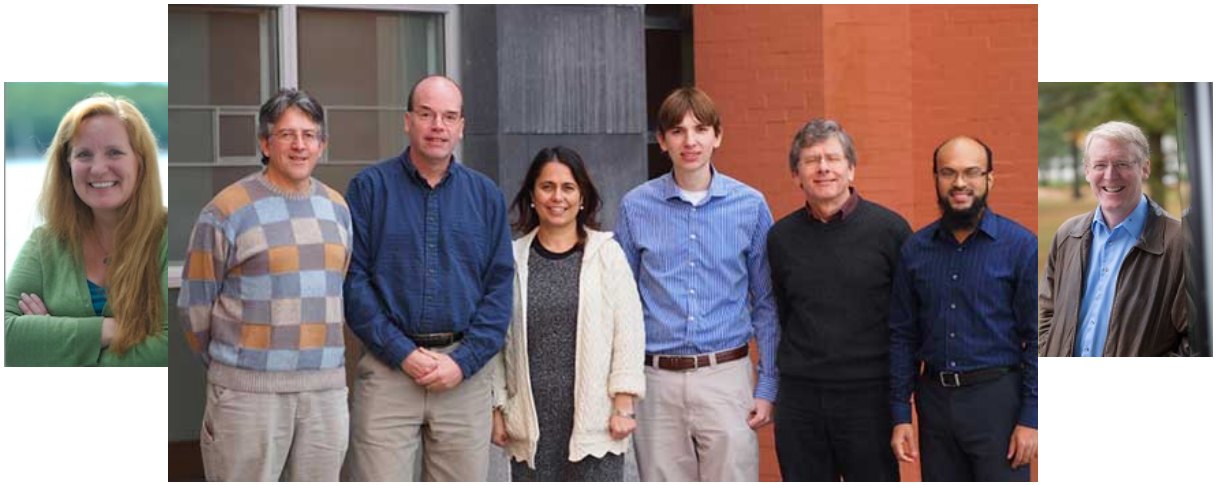
- Flow, flooding, sea-level rise, shortest path surfaces, visibility

- ..on very large data
- ..in parallel using Bowdoin's computing grid

# Summer 2016 Research

- Xx students
  - funded by institutional and faculty grants
- Example projects

# Computer Science



Location:  Searles, 2nd floor

Come talk to us!

# CSCI 1101
# Introduction to Computer Science

Introduction to problem solving and algorithmic thinking using computer programming. Provides tools and skills that can be used in any discipline. (Note: class is required for CSCI majors and minors, unless they place out).

## Topics:

- Problem solving
- Algorithm design
- Fundamentals of programming

## High-level questions:

- How do we design an algorithm to solve a problem?
- What kinds of problems can we solve with an algorithm?
- How can we use a computer to code and run an algorithm?

## Example activities:

- Build interactive games like Pong and 2048
- Animate scenes and pictures
- Encrypt text messages
- Build a spell checker

## Technology used:

- Programming in Java (or Python)